What is claimed is:

1. A method of profiling a threaded program during program runtime, the method comprising:

monitoring information exchanged between a processing unit and first and second threads executed by the processing unit;

determining, based on the information exchanged between the processing unit and the first and second threads, a critical path of thread execution and maintaining the critical path of thread execution in a critical path tree;

determining, based on the information exchanged between the processing unit and the first and second threads, a wait time during which the first thread awaits a synchronization event; and

determining whether the wait time affects the critical path of thread execution.

2. A method as defined by claim 1, further comprising indicating that the wait time is of a high priority if the wait time affects the critical path of thread execution and indicating that the wait time is of a low priority if the wait time does not affect the critical path of thread execution.

3. A method as defined by claim 1, wherein a leaf is added to the critical path tree when the synchronization event is a fork event.

4. A method as defined by claim 1, wherein a leaf is added to the critical path tree when the synchronization event is a signal event.

5. A method as defined by claim 1, wherein a leaf is removed from the critical path tree when the synchronization event is a wait event.

6. A method as defined by claim 1, wherein a leaf is added to the critical path tree when the synchronization event is an entry event.

7. A method as defined by claim 1, wherein a leaf is removed from the critical path tree when the synchronization event is a block event.

8. A method as defined by claim 1, wherein a leaf is removed from the critical path tree when the synchronization event is a suspend event.

9. A method as defined by claim 1, wherein a leaf is added to the critical path tree when the synchronization event is a resume event.

10. A method as defined by claim 1, further comprising comparing a number of active threads to a number of processing resources to determine a utilization factor.

11. An article of manufacture comprising a machine-accessible medium having a plurality of machine-accessible instructions that, when executed, causes a machine to:

monitor information exchanged between a processing unit and first and second threads executed by the processing unit;

determine, based on the information exchanged between the processing unit and the first and second threads, a critical path of thread execution and maintaining the critical path of thread execution in a critical path tree;

determine, based on the information exchanged between the processing unit and the first and second threads, a wait time during which the first thread awaits a synchronization event; and

determine whether the wait time affects the critical path of thread execution.

12. A machine-accessible medium as defined by claim 11, wherein the plurality of machine-accessible instructions, when executed, causes a machine to indicate that the wait time is of a high priority if the wait time affects the critical path of thread execution and indicating that the wait time is of a low priority if the wait time does not affect the critical path of thread execution.

13. A machine-accessible medium as defined by claim 12, wherein the plurality of machine-accessible instructions, when executed, causes a machine to add a leaf to the critical path tree when the synchronization event is a fork event.

14. A machine-accessible medium as defined by claim 12, wherein the plurality of machine-accessible instructions, when executed, causes a machine to add a leaf to the critical path tree when the synchronization event is a signal event.

15. A machine-accessible medium as defined by claim 12, wherein the plurality of machine-accessible instructions, when executed, causes a machine to remove a leaf from the critical path tree when the synchronization event is a wait event.

16. A machine-accessible medium as defined by claim 12, wherein the plurality of machine-accessible instructions, when executed, causes a machine to add a leaf to the critical path tree when the synchronization event is an entry event.

17. A machine-accessible medium as defined by claim 12, wherein the plurality of machine-accessible instructions, when executed, causes a machine to remove a leaf from the critical path tree when the synchronization event is a block event.

18. A machine-accessible medium as defined by claim 12, wherein the plurality of machine-accessible instructions, when executed, causes a machine to remove a leaf from the critical path tree when the synchronization event is a suspend event.

19. A machine-accessible medium as defined by claim 12, wherein the plurality of machine-accessible instructions, when executed, causes a machine to add a leaf to the critical path tree when the synchronization event is a resume event.

20. A machine-accessible medium as defined by claim 12, wherein the plurality of machine-accessible instructions, when executed, causes a machine to compare a number of active threads to a number of processing resources to determine a utilization factor.

21. A method of profiling a threaded program during program runtime, the method comprising:

monitoring information exchanged between a processing unit and first and second threads executed by the processing unit;

determining when a cross-thread event has occurred;

determining, based on the cross-thread event, a critical path of thread execution and maintaining the critical path of thread execution in a critical path tree;

determining, based on the cross-thread event and the information exchanged between the processing unit and the first and second threads, a wait time during which the first thread awaits a synchronization event; and

determining whether the wait time affects the critical path of thread execution.

22. A method as defined by claim 21, further comprising indicating that the wait time is of a high priority if the wait time affects the critical path of thread execution and indicating that the wait time is of a low priority if the wait time does not affect the critical path of thread execution..

23. A method as defined by claim 22, wherein the cross-thread event is selected from a group consisting of a fork event, an entry event, a signal event, a wait event, a suspend event, a resume event, and a block event.

24. A method as defined by claim 22, wherein a leaf is added to the critical path tree when the synchronization event is a fork event.

25. A method as defined by claim 22, wherein a leaf is added to the critical path tree when the synchronization event is a signal event.

26. A method as defined by claim 22, wherein a leaf is removed from the critical path tree when the synchronization event is a wait event.

27. A method as defined by claim 22, wherein a leaf is added to the critical path tree when the synchronization event is an entry event.

28. A method as defined by claim 22, wherein a leaf is removed from the critical path tree when the synchronization event is a block event.

29. A method as defined by claim 22, wherein a leaf is removed from the critical path tree when the synchronization event is a suspend event.

30. A method as defined by claim 22, wherein a leaf is added to the critical path tree when the synchronization event is a resume event.

31. A method as defined by claim 22, further comprising comparing a number of active threads to a number of processing resources to determine a utilization factor.

32. An article of manufacture comprising a machine-accessible medium having a plurality of machine-accessible instructions, when executed, causes a machine to:

monitor information exchanged between a processing unit and first and second threads executed by the processing unit;

determine when a cross-thread event has occurred;

determine, based on the cross-thread event, a critical path of thread execution and maintaining the critical path of thread execution in a critical path tree;

determine, based on the cross-thread event and the information exchanged between the processing unit and the first and second threads, a wait time during which the first thread awaits a synchronization event; and

determine whether the wait time affects the critical path of thread execution.

33. A machine-accessible medium as defined by claim 32, wherein the plurality of machine-accessible instructions, when executed, causes a machine to indicate that the wait time is of a high priority if the wait time affects the critical path of thread execution and indicating that the wait time is of a low priority if the wait time does not affect the critical path of thread execution.

34. A machine-accessible medium as defined by claim 32, wherein the plurality of machine-accessible instructions, when executed, causes a machine to add a leaf to the critical path tree when the synchronization event is a fork event.

35. A machine-accessible medium as defined by claim 32, wherein the plurality of machine-accessible instructions, when executed, causes a machine to add a leaf to the critical path tree when the synchronization event is a signal event.

36. A machine-accessible medium as defined by claim 32, wherein the plurality of machine-accessible instructions, when executed, causes a machine to remove a leaf from the critical path tree when the synchronization event is a wait event.

37. A machine-accessible medium as defined by claim 32, wherein the plurality of machine-accessible instructions, when executed, causes a machine to add a leaf to the critical path tree when the synchronization event is an entry event.

38. A machine-accessible medium as defined by claim 32, wherein the plurality of machine-accessible instructions, when executed, causes a machine to remove a leaf from the critical path tree when the synchronization event is a block event.

39. A machine-accessible medium as defined by claim 32, wherein the plurality of machine-accessible instructions, when executed, causes a machine to remove a leaf from the critical path tree when the synchronization event is a suspend event.

40. A machine-accessible medium as defined by claim 32, wherein the plurality of machine-accessible instructions, when executed, causes a machine to add a leaf to the critical path tree when the synchronization event is a resume event.

41. A machine-accessible medium as defined by claim 32, wherein the plurality of machine-accessible instructions, when executed, causes a machine to compare a number of active threads to a number of processing resources to determine a utilization factor.